

Introduction to MRI data processing with FSL

Anna Blazejewska

FSL = FMRI Software Library



- FMRIB = Functional Magnetic Resonance Imaging of the Brain @ Oxford
- since 2000, last stable FSL 5.0, **free!**
- for structural MRI, functional MRI (task, resting), diffusion MRI
- data processing & analysis

- written in C++ & TCL for Linux (virtual box on Windows) & Mac OS
- GUI but also **command line!!** → shell script pipelines
- installation: `fsinstaller.py`
- support: wiki, FAQ, forum

- yearly FSL courses: theory & hands on, 5 days, slides on-line
<http://fsl.fmrib.ox.ac.uk/fslcourse/>

FSL overview

structural MRI

- registration: linear (FLIRT) & non-linear (FNIRT)
- brain segmentation/extraction (BET)
- tissue type segmentation (FAST)
- subcortical structures segmentation (FIRST)
- voxelwise GM density analysis (FSLVBM)
- atrophy estimation (SIENA)

functional MRI

- motion correction (MCFLIRT)
- EPI distortion correction (FUGUE, PRELUDE)
- model-based analysis (FEAT)
- model free ICA-based analysis (MELODIC)
- Bayesian analysis of perfusion, ASL data (FABBER, VERBENA, BASIL)

diffusion MRI

- distortion correction (TOPUP)
- eddy current correction (EDDY)
- diffusion toolbox (FDT)
- tract-based spatial statistics (TBSS)
- general: fslmaths, fsfchfiletype, fslroi, smoothing, stats, fslview and more...

FSL overview

structural MRI

- registration: linear (FLIRT) & non-linear (FNIRT)
- brain segmentation/extraction (BET)
- tissue type segmentation (FAST)
- subcortical structures segmentation (FIRST)
- voxelwise GM density analysis (FSLVBM)
- atrophy estimation (SIENA)

functional MRI

- motion correction (MCFLIRT)
- EPI distortion correction (FUGUE, PRELUDE)
- model-based analysis (FEAT)
- model free ICA-based analysis (MELODIC)
- Bayesian analysis of perfusion, ASL data (FABBER, VERBENA, BASIL)

diffusion MRI

- distortion correction (TOPUP)
- eddy current correction (EDDY)
- diffusion toolbox (FDT)
- tract-based spatial statistics (TBSS)
- general: fslmaths, fslchfiletype, fslroi, smoothing, stats, fslview and more...

1. segmentation
2. registration
3. distortion correction
4. shell scripts!

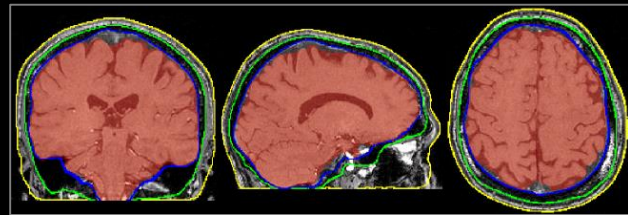
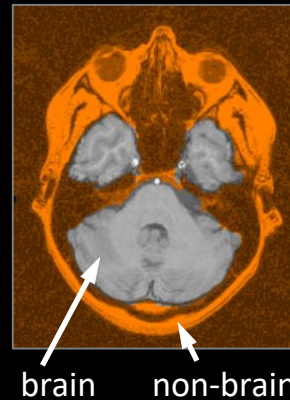


```
$ fsl
```

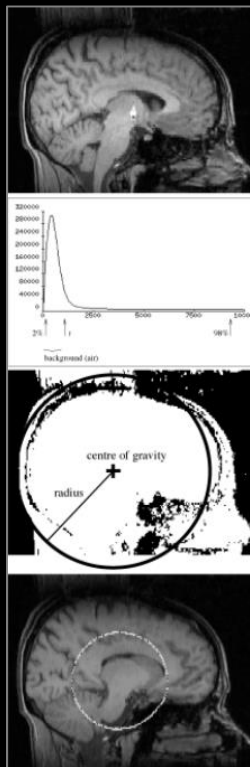
1. segmentation

BET: brain extraction = skull stripping

- goal: automatic segmentation of brain & non-brain tissue (skin, skull, eyeballs, etc.)
- preparation for:
 - registration/motion correction
 - tissue segmentation
 - masking out non-brain
- problems of manual segmentation: time, training, reproducibility
- benefits of **BET**
 - 5-20 sec, high reproducibility
 - different contrasts T1-/T2-/T2*-w data,
 - robust to bias field (uses local intensity changes)
 - can estimate inner & outer skull & outer scalp surface



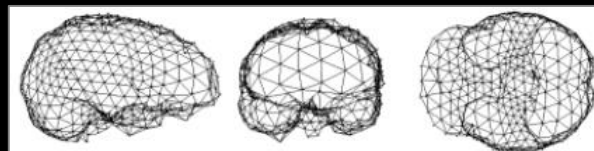
BET: how does it work?



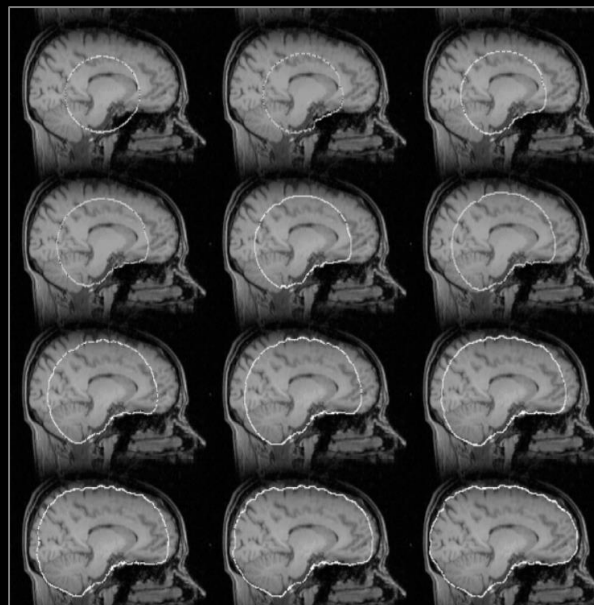
histogram-based
threshold t estimation

t -based binarization
center of gravity (COG)

spherical surface
initialization



surface = mesh
of connected triangles
(no folds, not cortex)



subdividing each triangle
expanding the surface

vertex locations updated
based on local intensities

not self-intersecting

surface smoothness condition

Smith, SM, *Fast robust automated brain extraction*, HBM 17(3), 2002.

BET: how to use it?

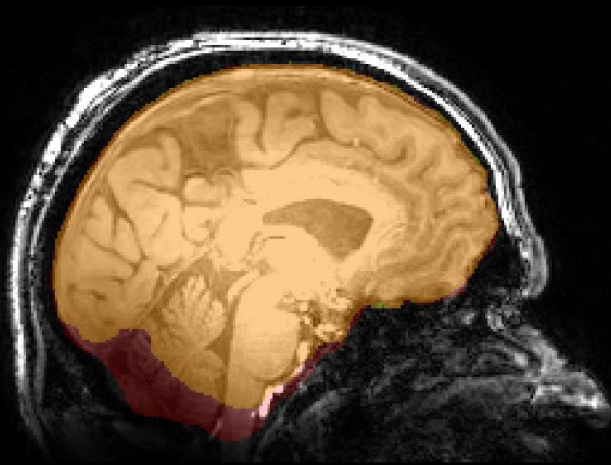
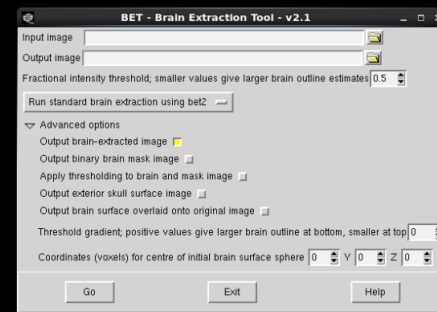
```
$ bet input output [parameters]
```

-f <f>

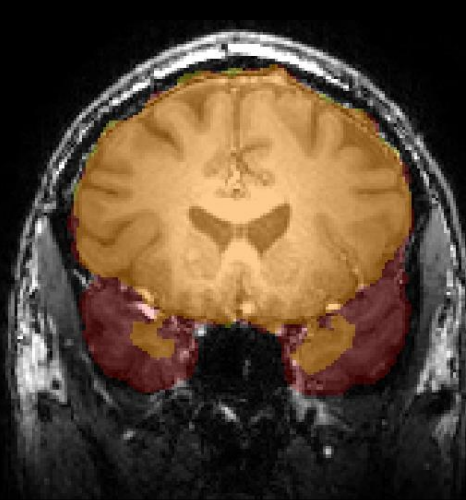
↓f → ↑brain outline estimate, [0,1], =0.5

-g <g>

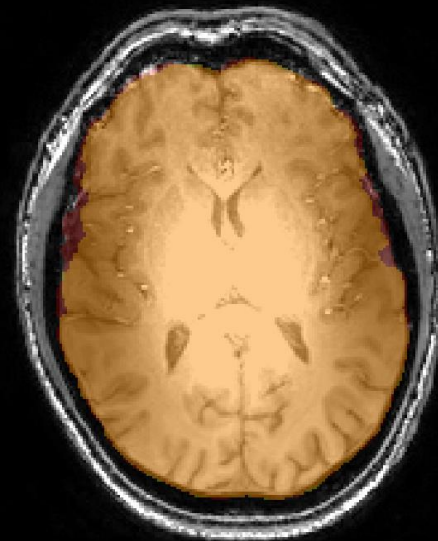
↑g → brain outline ↑bottom, ↓top, [-1,1], =0



7T Philips MPRAGE 1.0 mm³



default: f=0.5 g=0 modified: f=0.3 g=0.3



BET: how to use it?

```
$ bet input output [parameters]
```

```
-c <x y z>  initial center of gravity [voxels]  
-r <r>      head radius [mm], initial r/2
```

```
-R  robust brain center estimation  
-Z  temporary add slices for small z-FOV  
-F  apply to 4D fMRI data (-f 0.3 + dilation)
```

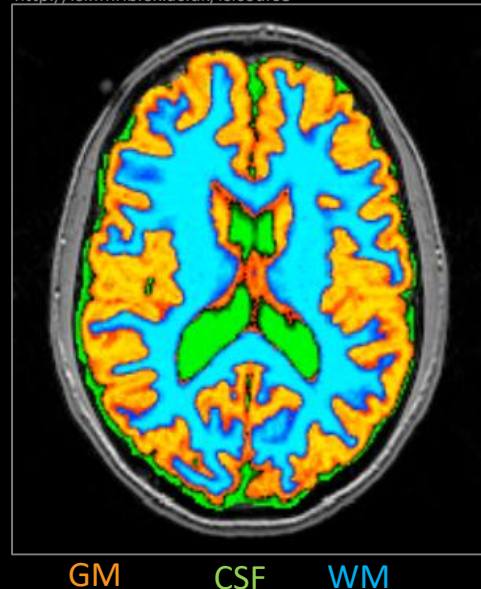
```
-m  save binary brain mask  
-s  save skull image  
-o  save brain surface outline  
-n  skip segmented brain image  
-A  generate skull & scalp surfaces (BETSURF)
```

FAST: tissue segmentation

FMRIB's Automated Segmentation Tool

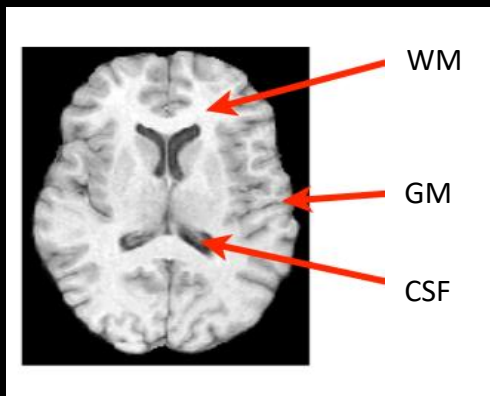
- goal: automatic segmentation of WM, GM & CSF
 - input:
 - BET processed images (extracted brain)
 - single image: T1, T2, PD
 - multichannel images pre-aligned (with FLIRT)
 - output: binary tissue masks or probability maps
1. histogram-based: uses Gaussian mixture model
 2. estimates & removes bias field
 3. considers voxel's neighborhood → robust to noise
 4. use of prior tissue probability maps

<http://fsl.fmrib.ox.ac.uk/fslcourse>

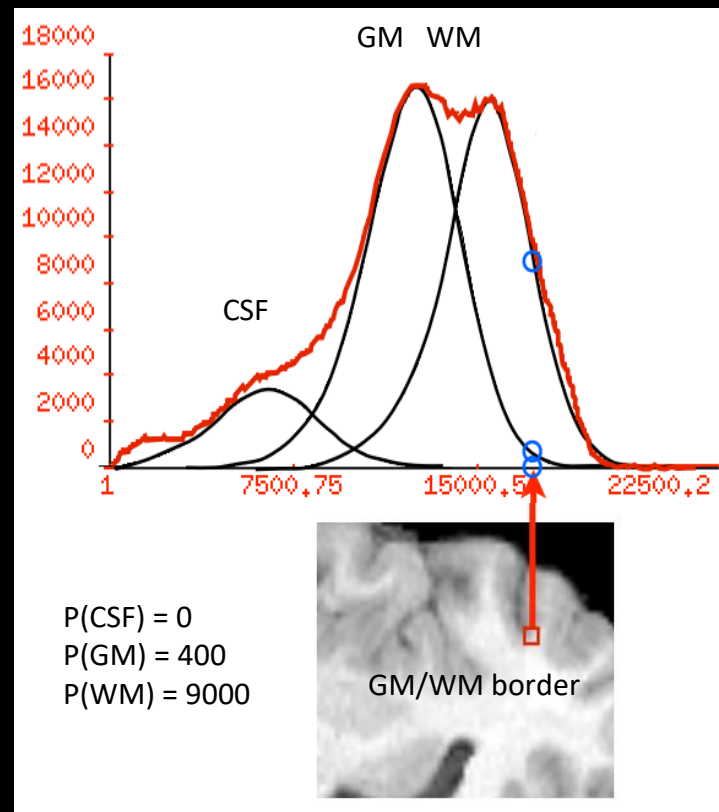


FAST: histogram as a mixture of Gaussians

- data → **intensity histogram**
- model = **mixture of Gaussians**
- separation of the peaks → segmentation
- peaks overlap → segmentation more difficult
- for each voxel calculates: $P(\text{CSF})$, $P(\text{GM})$, $P(\text{WM})$

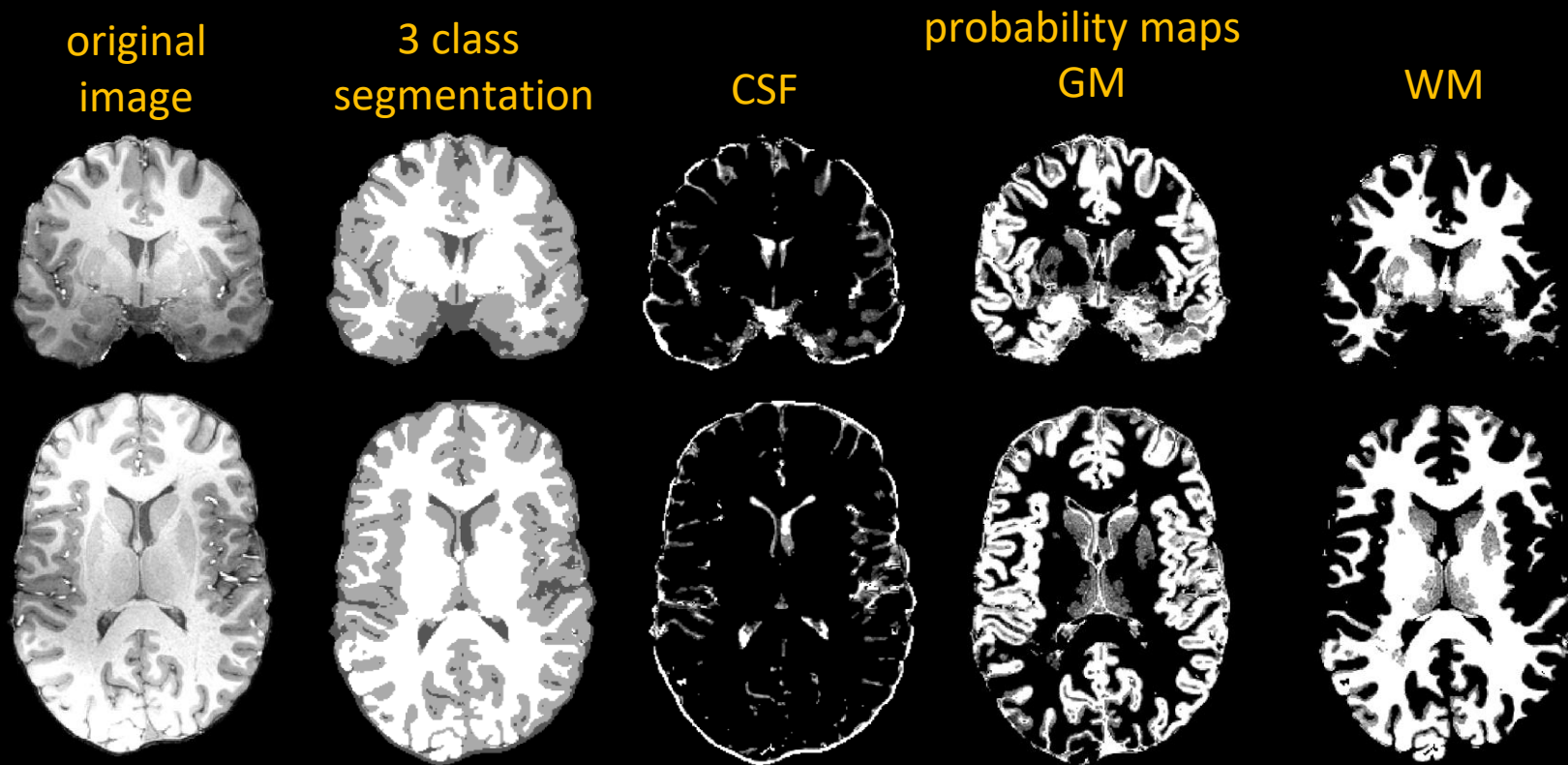


<http://fsl.fmrib.ox.ac.uk/fslcourse>



<http://fsl.fmrib.ox.ac.uk/fslcourse>

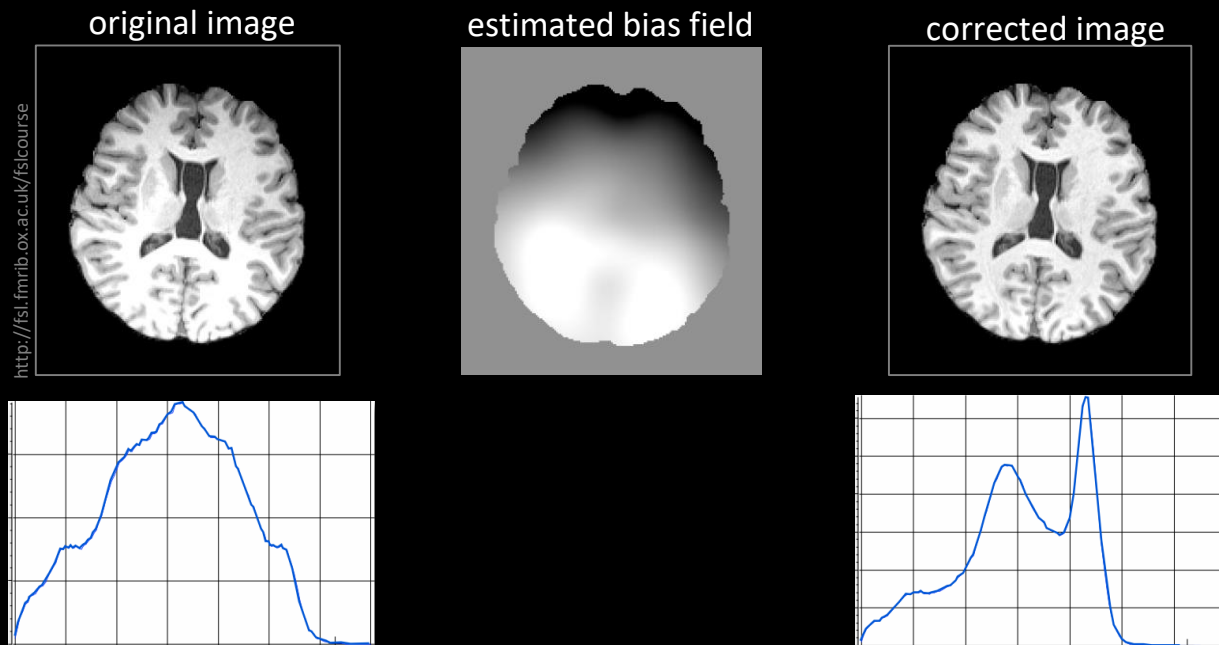
FAST: hard segmentation vs probability maps



7T Siemens MEMPRAGE 0.75 mm³

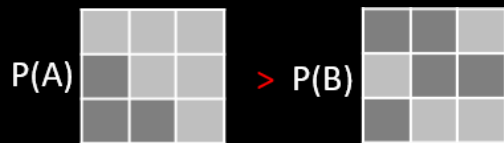
FAST: bias field correction

- low resolution, head motion, noise, blurring & **bias field** → histogram peaks overlap
- RF inhomogeneity → spatial intensity variations = **bias field**

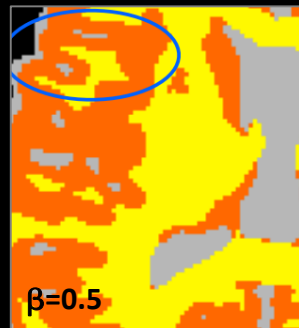
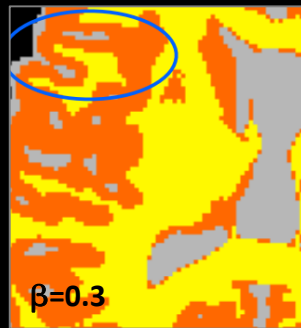
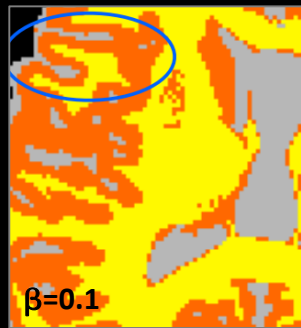
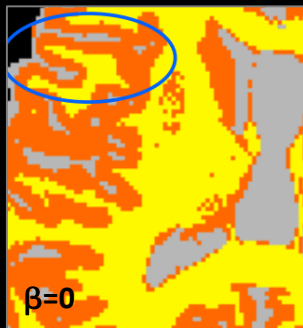


FAST: neighborhood

- local neighborhood information → robust to noise



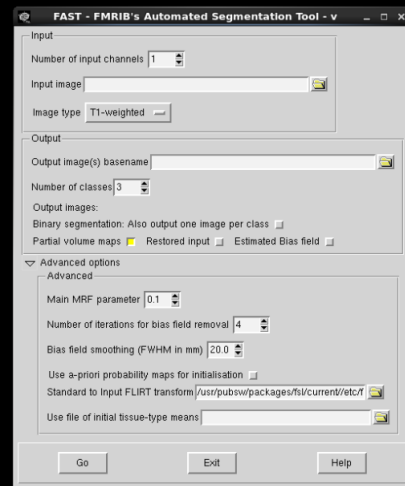
- $P(\text{class}) = P(\text{intensity}) + \beta P(\text{neighborhood})$
- β controls contribution of neighbors vs intensity, can be set by user



FAST: how to use it?

```
$ fast [options] input
```

- H** $\beta, =0.3$
- t** input data type: 1=T1, 2=T2, 3=PD
- S** number of input data channels, =1
- b** save estimated bias field
- B** save bias-corrected input image
- N** no bias field correction
- I** number of iteration for bias field removal, =4
- l** bias field smoothing FWHM, =20mm
- n** number of tissue type classes, =3
- g** save a binary mask for each class
- nopve** skip probability maps



FAST: use of priors

```
-a t.mat
```

use prior probability maps for initialization

(requires transformation to standard space)

```
-P
```

use prior probability maps at all stages (requires -a/-A)

```
-A p1 p2 p3
```

alternative prior images for tissue classes

```
-s file
```

initial tissue-type means

T1-w mean



GM



WM



CSF

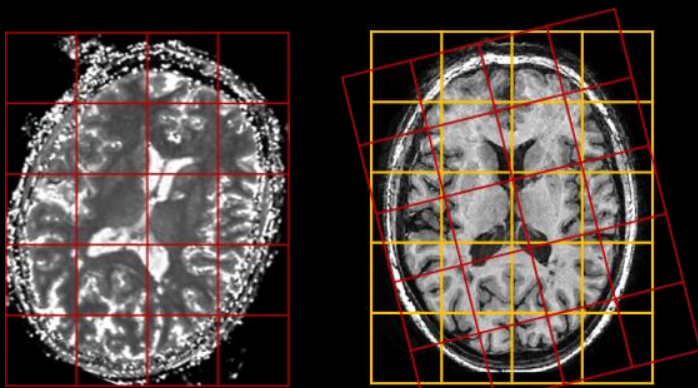


<http://fsl.fmrib.ox.ac.uk/fslcourse>

2. registration

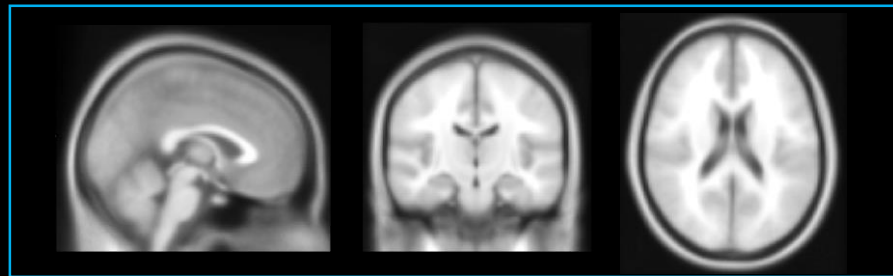
registration

- = transformation of 3D volumes into the same space (coordinate system)
- 1. transformation 2. cost function 3. interpolation
- **intra-subject**: different modalities, contrasts, frames etc.
- **inter-subject**: different subjects to common space (population analysis etc.)



7T Philips T1 map & MPRAGE 0.6 mm³

MNI152 = 152 structural images, averaged after high-dimensional nonlinear registration

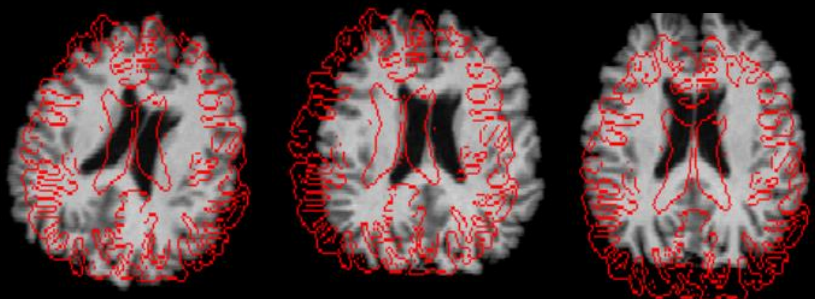


MNI152lin_T1_1mm @ \$FSLDIR/data/standards

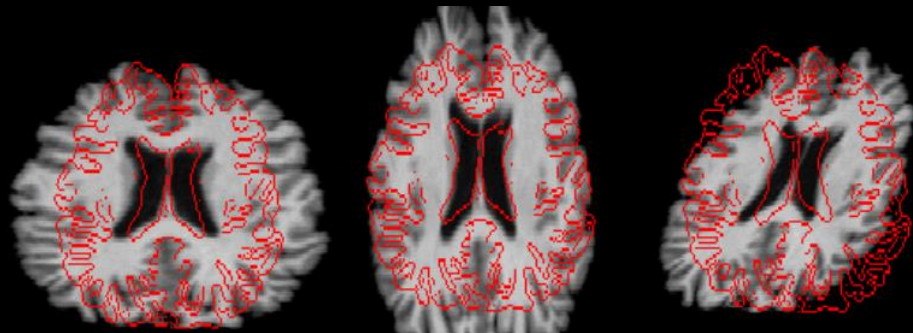
FLIRT: linear transformation

FMRIB's Linear Registration Tool

- intra-subject, same/different modality, DOF = **degrees of freedom**



rigid body = **6DOF** = 3 rotations + 3 translations



affine = **12 DOF** = 6DOF + 3 scalings + 3 skews/shears

- represented by a **transformation matrix**

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

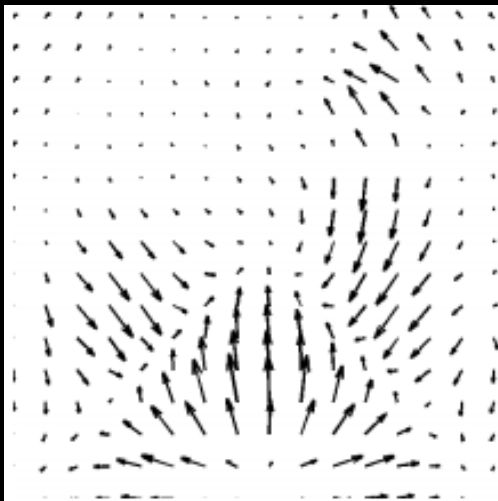
- coordinate vectors multiplied by the matrix define transformed coordinates

FNIRT: non-linear transformation

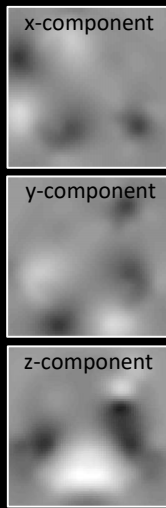
FMRIB's Non-Linear Registration Tool

- inter-subject: same anatomical region in all subjects (atlas with anatomical labels)
- many many DOFs, same modality, least squares cost function, bias field model (FNIRT)
- represented by a 3D **deformation field** = warp, displacement field stored as 3 x 3D volumes (x-, y-, z-components of the vectors)

deformation field

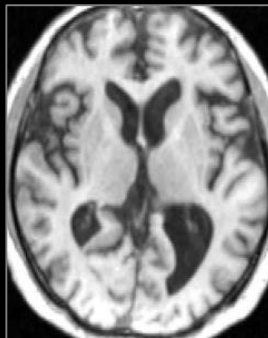


<http://fsl.fmrib.ox.ac.uk/fslcourse>



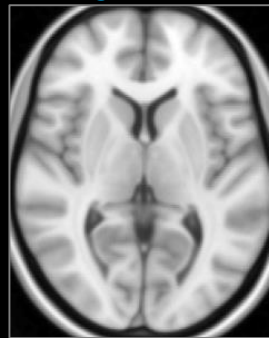
-6 6 mm

FLIRT

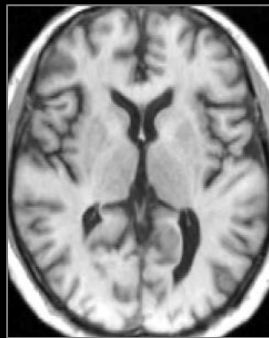


<https://fsl.fmrib.ox.ac.uk/fsl/fslwiki>

MNI152



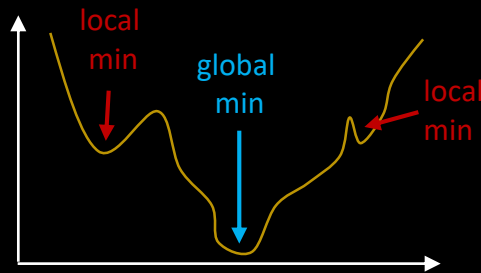
FNIRT



! inter-subject variability – not very accurate
! some individual features lost

cost function

- registration = iterative process of finding the best alignment of 3D volumes
- minimizing cost function = measure of goodness of the alignment
- beware of local minima!

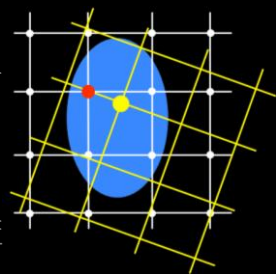


- **least squares**: same modality, same brightness, contrast (includes bias field model)
- **normalized correlation**: same modality, different brightness, contrast
- **correlation ratio**: any MRI modalities
- **mutual information**: any modalities (also CT, PET etc.)
- **normalized mutual information**: any modalities (also CT, PET etc.)
- **boundary-based-registration = BBR**: intra-subject, EPI to structural

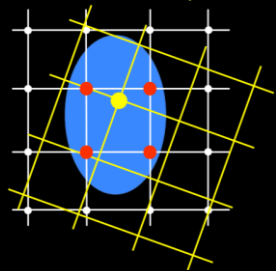
interpolation

=finding intensity values between grid points

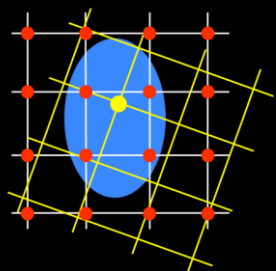
<http://fsl.fmrib.ox.ac.uk/fslcourse>



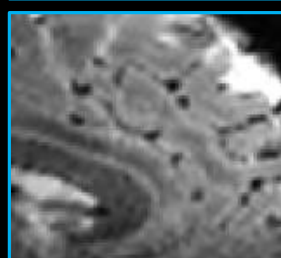
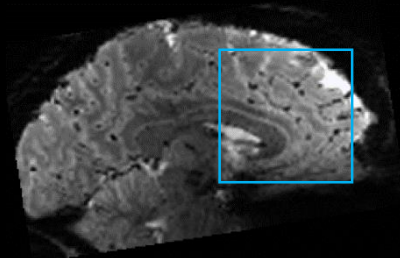
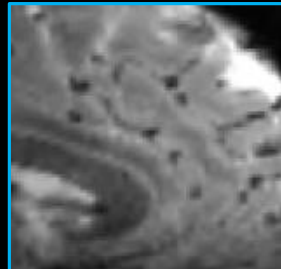
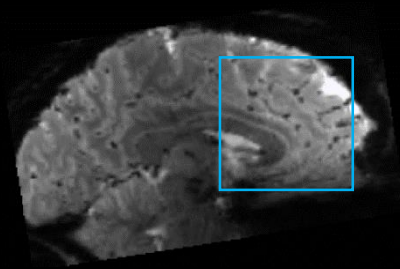
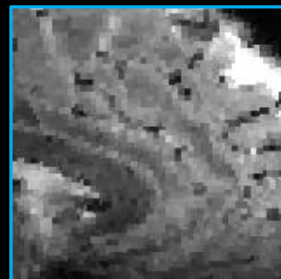
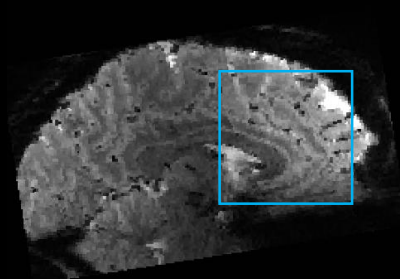
nearest neighbor (NN)
fast, blocky (pixelized)
for labels, binary masks



tri linear (TL)
fast, blurring
most common option



spline, sinc
slower, sharper images
(spline not that slow)



7T Siemens GRE EPI 1.5 mm³

FLIRT: how to use it?

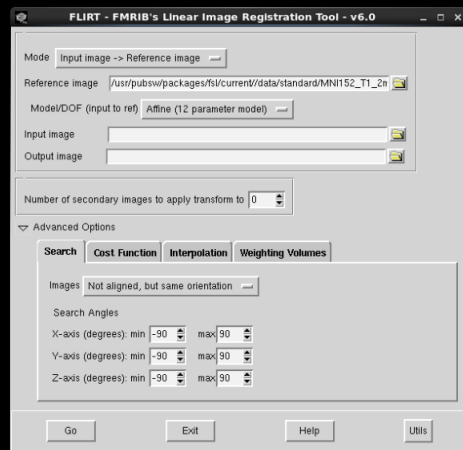
```
$ flirt -in input -ref reference  
-out output -omat matrix
```

```
-dof df  
-cost cf  
-interp it
```

degrees of freedom, =12
cost function, =corratio
interpolation method, =trilinear

```
-searchrx min max  
-nosearch  
-usesqform  
-noresample
```

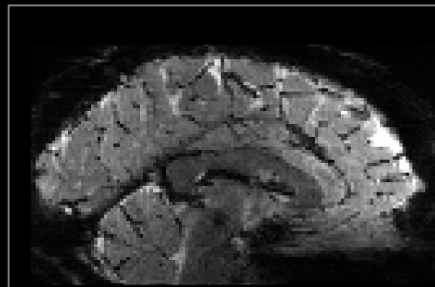
search angles, =-90 90
all search angles = 0
initialize using sform/qform
estimates transformation



FLIRT: example



7T Siemens MEMPRAGE 0.75 mm³



7T Siemens GRE EPI 1.5 mm³



```
$ flirt -in input -ref reference -out output -omat matrix  
-cost mutualinfo -dof 6 -interp sinc  
-searchrx -3 3 -searchry -3 3 -searchrz -3 3
```


FLIRT: mask registration

- applying existing transformation:

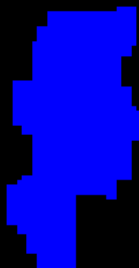
```
$ flirt -in input -ref reference -out output  
-applyxfm -init matrix.mat
```

`-interp nearestneighbour` integer, rounded \rightarrow mask shrinks

`-interp trilinear` followed by thresholding: $= 0.5 \approx$ same size mask as input
 < 0.5 more PVE, > 0.5 less PVE



original



NN



TL



TL vs NN



TL 0.5 vs NN



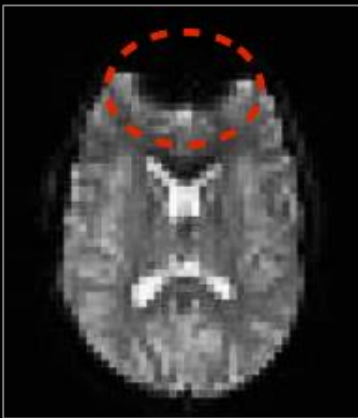
TL 0.2 vs NN

FLIRT & FNIRT: weighted registration

- cost function weighting for input and/or reference image
values range [0,1], ↓ for corrupted areas, ↑ for important areas

artifacts

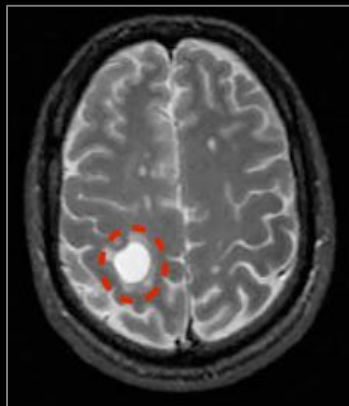
<http://fsl.fmrib.ox.ac.uk/fslcourse>



=1

=0

pathologies



`-refweight <volume>`

`-inweight <volume>`

use weights for reference volume

use weights for input volume

MCFLIRT: motion correction

- registration of multiple frames of the same fMRI scan

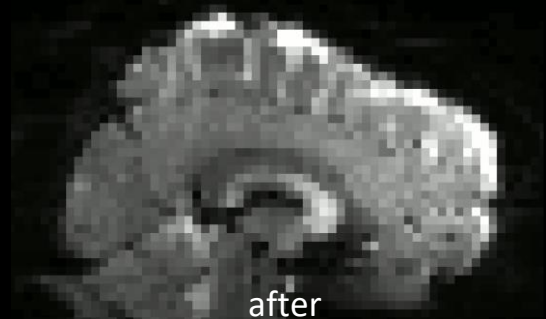
```
$ mcflirt -in input -out output [options]
```

-refvol reference frame, =middle
-meanvol use mean volume as a reference

-cost cf cost function, =**normcorr**
-dof df degrees of freedom, =6

-sinc_final
-spline_final final interpolation choice, =trilinear

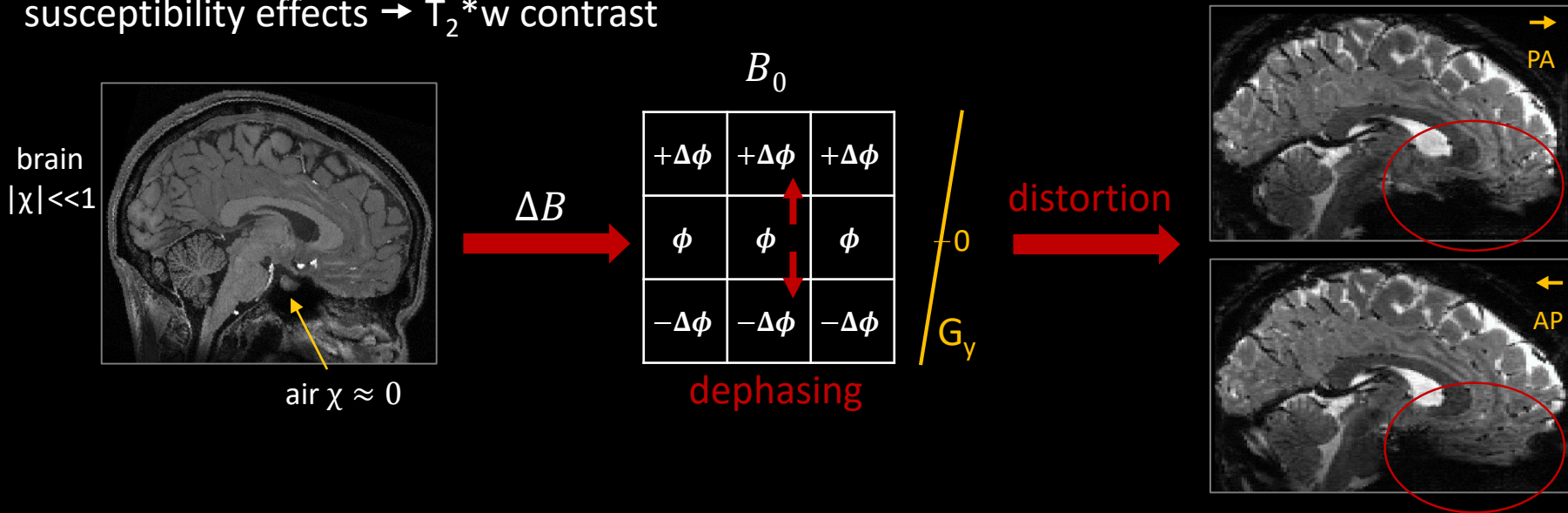
-stats save variance & stdev images
-mats save motion parameters in*.par file
-plots save transformation matrices (in subdir)



3. distortion correction

EPI geometric distortion

magnetic susceptibility = χ material's property, quantifies ability to magnetize
 susceptibility effects $\rightarrow T_2^*$ w contrast



1. field map based distortion correction (FUGUE)
2. EPI data acquired with the opposite PE direction (TOPUP)

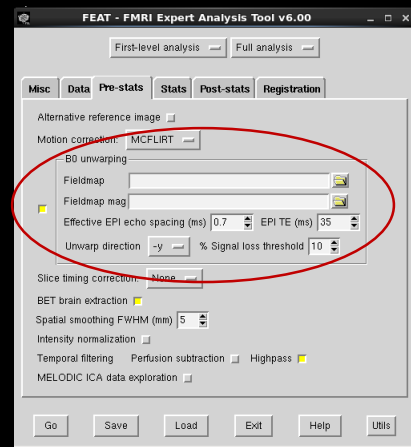
FUGUE: how to use it?

- field map $\rightarrow \Delta B \rightarrow$ voxel shift map \rightarrow undo distortion (unwarp)
- need to have/know:
field map, phase encoding direction, ESP, TEs (ΔTE)

```
$ fugue -i epi -p phase -u output
```

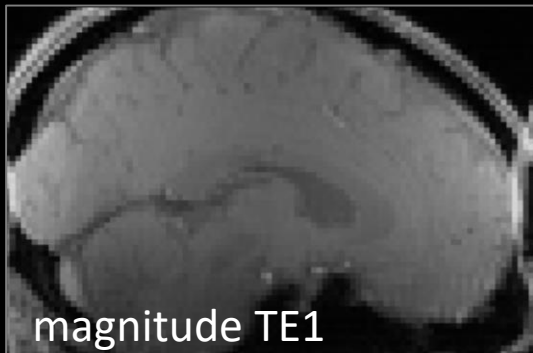
```
--unwarpcdir=dir  
--dwell=ESP  
--asym=dTE  
-s sigma
```

phase encoding direction, =y (usually AP)
echo spacing (ESP, in sequence parameters)
 ΔTE
2D Gaussian smoothing (field map regularization)

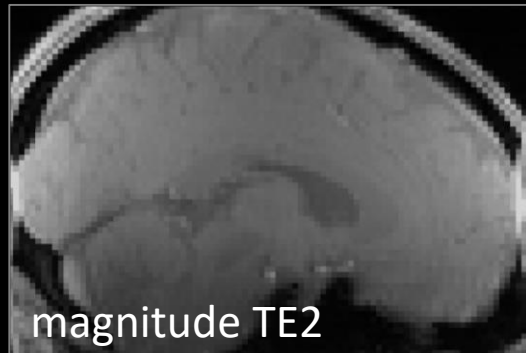


! phase data must have the same FOV & resolution as EPI (may require prior resampling)

FUGUE: how to use it?



magnitude TE1



magnitude TE2



Δ phase

7T Siemens, 2mm³

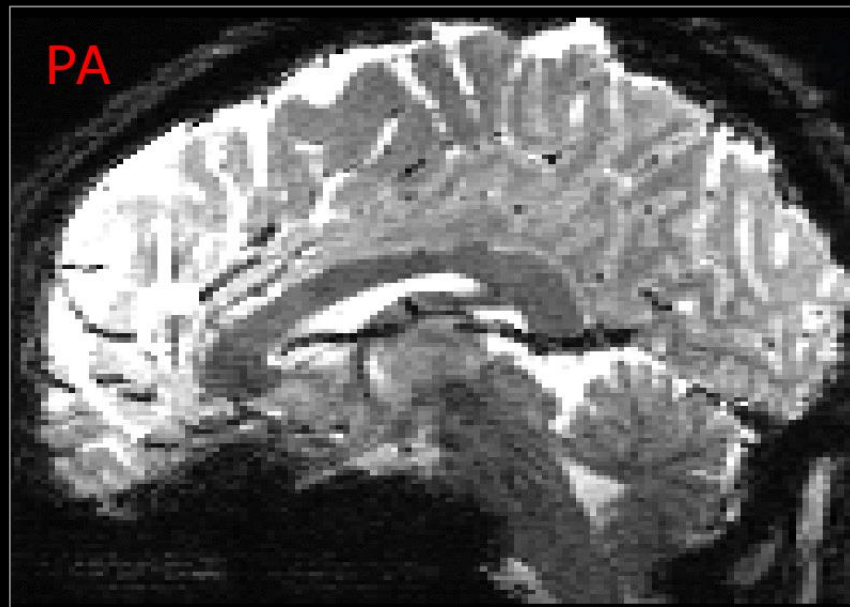
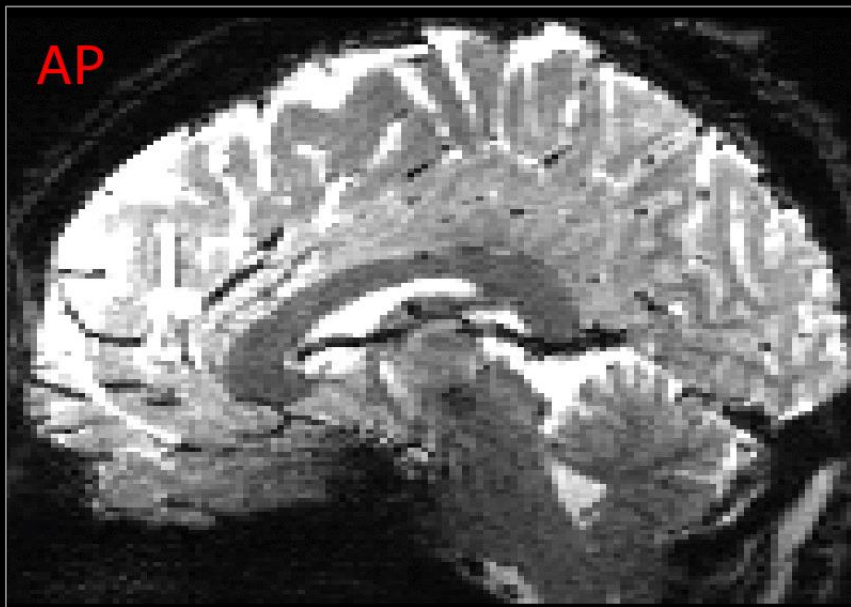
- Siemens: GUI/script, magnitude data brain only (BET), output in rad/s

```
$ fsl_prepare_fieldmap SIEMENS phz mag output dTE
```

- others (or Siemens) manually:
 - resampling to EPI resolution: `flirt -applyxfm`
 - convert phase to radians & rad/s: `fslmaths`
 - unwrapping phase: `prelude`

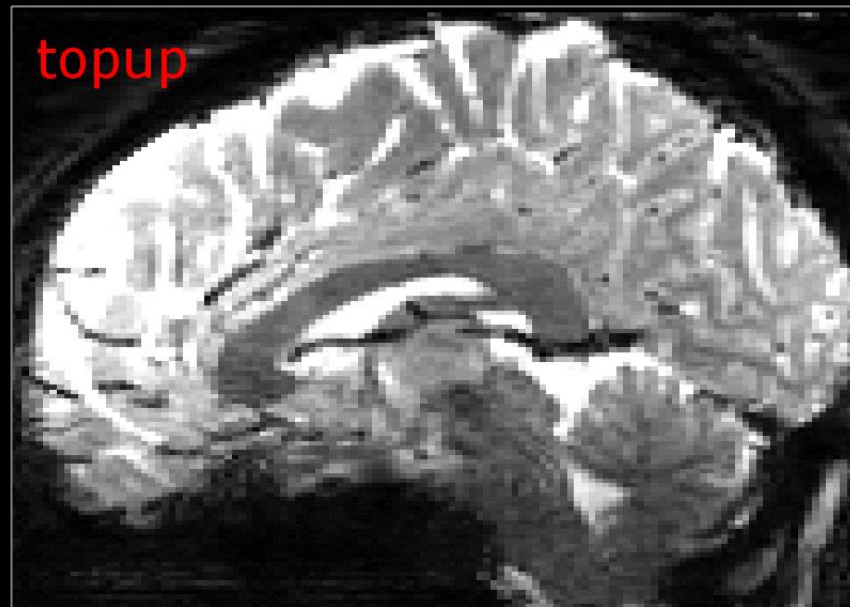
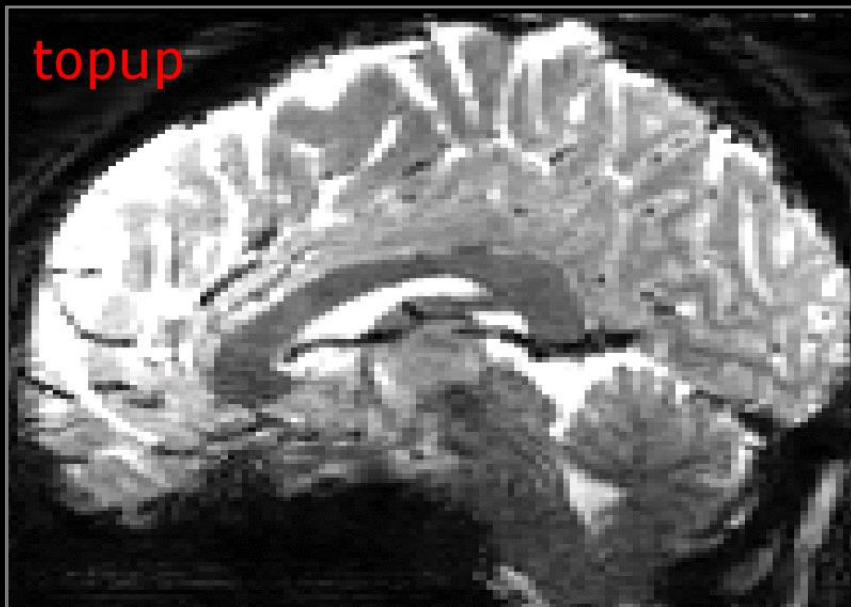
TOPUP: distortion correction

- EPI data acquired with opposite phase encoding directions



TOPUP: distortion correction

- EPI data acquired with opposite phase encoding directions



TOPUP: how to use it?

```
$ topup --imain=input --datain=ap_pa.txt --config=topup.cfg  
      --out=output --fout=fmap  
      --iout=corrected --dfout=deform
```

- input: combined frames with opposite phase encoding directions
- ap_pa.txt:
columns 1-3 = phase encoding directions (3rd one must be 0),
column 4 = readout time (if vary between the volumes)
- config: default one `$FSLDIR/etc/flirtsch/b02b0.cnf`
(data resolution <3.0 mm)
- corrected: input set of frames
otherwise `applytopup`

ap_pa.txt

0	1	0	1
0	1	0	1
0	1	0	1
0	-1	0	1
0	-1	0	1
0	-1	0	1

shell scripting

```
REF=${PATH_TO_ALL_MY_DATA}/struct_brain.nii.gz
for s in ${PATH_TO_ALL_MY_DATA}/epi*brain.nii.gz; do
    flirt -in ${s} -ref ${REF} -dof 6 -cost mutualinfo...
done;
```

- automatic! work by themselves!
- run multiple subjects/datasets automatically
- systematically explore different parameters/options
- run multiple processing streams in parallel
- reproducibility: each subjects processed the same way
- replicability: easy to re-run with some modification
- save (are) documentation of what was run
- easy to share with others

! check the results on each stage of processing & monitor potential crashing..

Thank you!

Anna Blazejewska

ablazejewska@mgh.harvard.edu